

PROPOSAL

Implementation of PACKAGE in Compis-Comal.

=====

A substantial increase in the possibility to implement new functions in Comal can be achieved by introducing PACKAGES. This is also a way to have new functions without conflict with the content of the Comal KERNEL. Packages in this context means the possibility to connect functions/procedures written in other languages than Comal. Furthermore the package code module called from Comal is in machine code and will execute immediately as opposed to the Comal program that is interpreted. The package functions/procedures are completely independent with the only restriction that parameters are handled in a standardized manner.

Comal today will allow the calling of INTERNAL and EXTERNAL functions/procedures. They must be written in Comal. If external procedures/functions are used, the program execution is quite slow, especially if the hardware does not have hard- and/or RAMdisc. That is because Comal has to read external proc/func from disc. These disadvantages will be eliminated if we introduce a new type of proc/funcs - PACKAGE. It is suggested that the declaration of package proc/funcs is exactly the same as for external func/procs namely:

```
PROC      procname [(par [,par]...)] PACKAGE "packagename"
FUNC      funcname [(par [,par]...)] PACKAGE "packagename"

par       [REF] 'nvar'
par       [REF] 'svar'
par       REF 'ntab' ([,]...)
par       REF 'stab' ([,]...)
```

A considerable advantage is that the model of package implementation described above closely follows the way Comal calls external proc/funcs and consequently a large part of that mechanism already is a part of Comal.

To make it easy to work with packages we suggest that the Comal environment is changed/upgraded as follows.

- a) Line numbers reserved for package declarations.
- b) Package declarations are stored in the last part of line numbers.
- c) LINK - Command to load packages.
- d) DISCARD - Command to throw away packages.
- e) MAKE - Command to create and check packages.
- f) LIST - the list command is extended to handle packages.
- g) LISTPACK - Command to toggle the list/nolist of package declarations

DESCRIPTION OF CHANGES/IMPLEMENTATION.
=====

Introducing a mechanism that places all the package declarations at the end of the user program area. When loading packages there has to be a reserved field in the user program area that cannot be written to by the user, this field is placed at the end of accepted linenumbers.

CHANGING OF LIST so that program lines with package declarations will not be listed if not explicitly asked for.
When saving user programs on disk, no program lines with package declarations will be stored. This applies to both save and list files.

LISTPACK +/- (initial value -) Command

Listpack + is used to make LIST include lines with package declarations when listing device is console or printer.
When listing package declaration lines line numbers are substituted with ***** as is shown below:

```
***** PROC time1 (a, REF b(,), c) PACKAGE "control"  
***** FUNC time2 (a,b) PACKAGE "control"
```

LINK "filename" [, "packagename"] Command

Filename must have extension PAC. If packagename is specified it will substitute the original name of the package which is included in the package itself. When the command LINK is executed the following will happen:

- 1) The machine code module is loaded and that part of memory is allocated.
- 2) Corresponding package proc/func declarations, written in Comal, are written into the user program memory of Comal starting with line number 60000. The (eventually) following link commands there will be a concatenation of package declarations. A total of 1535 package declarations can be loaded
- 3) A table is created where Comal associates every proc/func name with a physical start address. The same table will be scanned during program editing/writing every time a program line with a package procedure call is written. Internal/external procedure names can also be included in this table. The consequence of this is that Comal will "remember" all called/declared procedures and there will be much less need for the question

Procedure call (J/N) (n)

when a program line including a call to a procedure is typed in.

DISCARD "packagename"

Command

DISCARD deallocates the part of the memory that has been occupied by the package "packagename". After that an automatic space compression takes place.

MAKE "filename" [,"packagename"]

Command

Make will create a packagefile from a .CMD-file and a Comal file. The Comal file must be stored as an ASCII file and can only contain package declarations and comments.

TeleLOGIC 840921

Kjell Andersson, Hakan Hjelmestam